2023012847 闫-畅

第一章

名词解释.

操作系统：管计算机硬件和软件的软件

实时操作系统：系统能及时响应外部事件的请求，在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行

互斥共享 某个资源可以给多个进程使用，但在一定时间内，只允许一个进程访问资源

填空.

1、硬件、软件 3、临界资源、共享资源. 5、共享性、异步性.

第二章

名词解释.

临界区：指进程中用于访问临界资源的代码

进程同步：是对多个相关进程在执行次序上进行协调，使并发执行的诸进程之间能按照一定的规则共享系统资源并能很好的相互合作，从而使程序的执行具有可再现性

填空.

1、PCB、动态性、异步性 5、动态、静态 6、~~阻塞~~资源竞争 8、共享内存、消息传递、管道通信

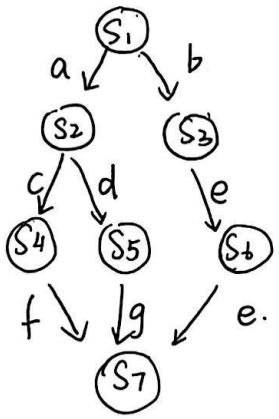
9、执行就绪、阻塞

问答题.

2. (1) 前驱关系：$S_1 \to S_2$ $S_1 \to S_3$ $S_2 \to S_4$ $S_2 \to S_5$ $S_3 \to S_6$.

~~$S_1 \to S_2 \to S_4 \to S_7$~~.

~~$S_1 \to$~~ $S_4 \to S_7$ $S_5 \to S_7$ $S_6 \to S_7$.



(2) ~~semaphore a,b,c,d,e,f,g=0,0,0,0,0,0,0~~

```
P1(){ S1; signal(a); signal(b); }
P2(){ wait(a); S2; signal(c); signal(d); }
P3(){ wait(b); S3; signal(e); }
P4(){ wait(c); S4; signal(f); }
P5(){ wait(d); S5; signal(g); }
P6(){ wait(e); S6; signal(e); }
P7(){ wait(f); wait(g); wait(e); S7; }
```

```
main(){
  semaphore a,b,c,d,e,f,g = 0,0,0,0,0,0,0;
  cobegin
    P1(); P2(); P3(); P4(); P5(); P6(); P7();
  coend
}
```

```
semaphore mutex=1, empty=5, orange=0, apple=0;
Dad(){ while(1){ wait(empty); wait(mutex);
                 将水果放入;
                 signal(mutex);
                 if(放入的是橘子) signal(orange);
                 else signal(apple); } }
```

```
Son(){while(1){wait(orange);
                wait(mutex);
                取出桔子
                signal(mutex);
                signal(empty);
                吃桔子;}}
```

```
Daughter(){while(1){wait(apple);
                    wait(mutex);
                    取出苹果;
                    signal(mutex);
                    signal(empty);
                    吃苹果;}}
```

4、

```
semaphore avail1=1, avail2=1.   Main(){cobegin
          full1=0   full2=0.                PA();PB();PC();
                                            coend}
PA(){while(1){从磁盘读入内存;
              P(avail1);
              放入缓冲区1;
              V(full1);}}
```

```
PB(){while(1)
     {P(full1);
      从缓冲区1取出;
      V(avail1);
      P(avail2);
      将记录放入缓冲区2;
      V(full2);}}
```

```
PC(){while(1)
     {P(full2);
      从缓冲区2中取出;
      V(avail2);
      打印记录;
     }}
```

```
int count=0
semaphore mutex=1 empty=1 full=0
semaphore payment=0 receipt=0
```

```
guest(){
  wait(mutex);
  if(count>=N)
    signal(mutex);
    离开理发店;
  }
  else{count++;
       signal(mutex);
       在沙发中就座;
       wait(empty);//等待理发椅变空.
       离开沙发,坐到理发椅上;
       wait(mutex);
       count--;
       signal(mutex);
       理发;
       signal(full);
       付费;
       signal(payment);//通知理发师付费
       wait(receipt);//等待理发师收费
       signal(empty);//离开理发椅
       离开理发店;}}
```

```
Barber(){
  while(1){
    wait(full);
    理发;
    wait(payment);
    收费;
    signal(receipt);
  }}
```

2. A松弛度：20-10=10ms. B松弛度：50-10=40ms C松弛度：50-15=35ms

A的松弛度最低，执行A、之后.

5

第3章

3

名词解释：

处理机调度：在多道程序环境下，内存中存在着多个进程，其数目往往多于处理机数目，要求系统能按某种算法，动态地将处理机分配给处于就绪状态的一个进程，使之执行.

周转时间：是指从作业被提交给系统开始，到作业完成为止的这段时间间隔。它包括四部分时间：作业在外存后备队列上等待（作业）调度的时间，进程在就绪队列上等待进程调度的时间，进程在CPU上执行的时间，以及进程等待I/O操作完成的时间

死锁：如果一组进程中的每一个进程都在等待仅由该组进程中的其它进程才能引发的事件.

填空：

2. 后备，运行和完成　　6. 预防死锁，避免死锁，检测死锁，解除死锁.

问答

1.

FCFS. 调度顺序 ⟹ A→B→C→D→E

①完成时间：A：3　B：3+6=9　C：9+4=13　D：13+5=18.　E：18+2=20

②周转时间：作业完成时间－作业到达时间

A：3-0=3　B：9-2=7　C：13-4=9　D：18-6=12　E：20-8=12

③带权周转时间：作业周转时间／作业运行时间

A：3/3=1　B：7/6≈1.167　C：9/4=2.25　D：12/5=2.4　E：12/2=6.

④平均周转时间　(3+7+9+12+12)÷5=8.6.

⑤平均带权周转时间　(1+1.167+2.25+2.4+6)÷5=2.5634

SJF.（非抢占式）调度顺序 A→B→E→C→D.

①完成时间　A：3.　B：9　E：11　C：15　D：20

②周转时间：A：3　B：7　E：3　C：11　D：14.

③带权周转时间

A：3/3=1　B：7/6≈1.167　E：3/2=1.5　C：11/4=2.75　D：14/5=2.8.

④平均周转时间：(3+7+3+11+14)÷5=7.6

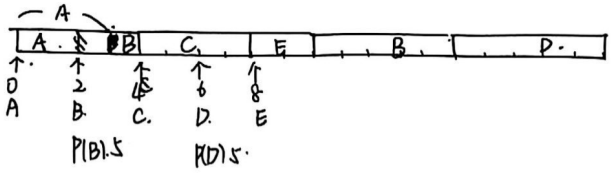
⑤平均带权周转时间：(1+1.167+1.5+2.75+2.8)÷5≈1.8434.

2.

A初始优先级：20-1=19

A=1

SJF（抢占式）



4

完成时间：A：3 B：15 C：8 E：10 D：20

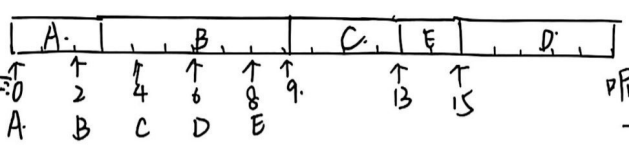
周转时间：A3-0=3 B：15-2=13 C：8-4=4 E：10-8=2 D：20-6=14.

带权周转时间：A：3/3=1 B：13/6=2.167 C：4/4=1 E：2/2=1 D：14÷5=2.8.

平均周转时间：(3+13+4+2+14)÷5=7.2

平均带权周转时间：(1+2.167+1+1+2.8)÷5=1.5934

HRRN.



响应比：B：$\frac{1+6}{6}=\frac{7}{6}$.

C：$\frac{5+4}{4}=2.25$.

D：$\frac{3+5}{5}=1.6$.

E：$\frac{1+2}{2}=1.5$. C>D>E

D：$\frac{7+5}{5}=2.4$

E：$\frac{5+2}{2}=3.5$ E>D.

完成时间：A：3 B：9 C：13 E：15 D：20.

周转时间：A3-0=3 B：9-2=7 C：13-4=9.

E：15-8=7 D：20-6=14

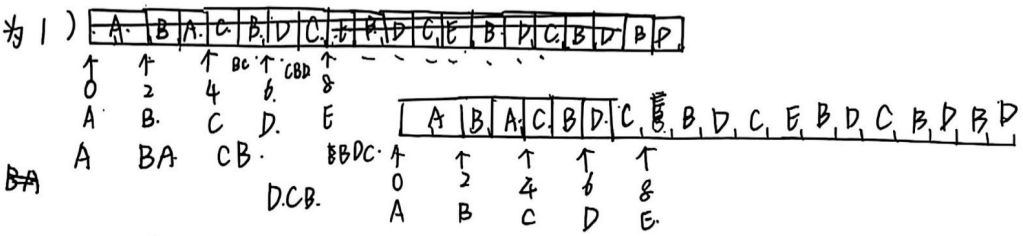
带权周转时间：A：3/3=1 B：7/6=1.167 C：9/4=2.25.

D：14/5=2.8 E：7/2=3.5.

平均周转时间：(3+7+9+7+14)÷5=8.

平均带权周转时间：(1+1.167+2.25+2.8+3.5)÷5=2.1434

RR（时间片为1）



完成时间：A：4. B：19 C：16. D：20 E：13.

周转时间：4-0=4 B：19-2=17 C：16-4=12 D：20-6=14 E：13-8=5

带权周转时间：A：4/3=1.33 B：17÷6=2.83 C：12÷4=3 D：14÷5=2.8 E：5/2=2.5

平均周转时间：(4+17+12+14+5)=10.4

平均带权周转时间：(1.33+2.83+3+2.8+2.5)÷5=2.492

2. A松弛度：20-10=10ms. B松弛度：50-10=40ms C松弛度：50-15=35ms

A的松弛度最低，执行A，之后

B：40-10=30ms C：35-10=25ms

C的松弛度最低，执行C，之后求执行B.

序：A→C→B

3.

Process	Work	Need	Allocation	work+allocation
P_0	1622	~~0044~~ 0012	~~0044~~ 0032	1654 ✓
P_3	1654	~~1750~~ 0652	0332	1986
P_4	1986	0656	0014	19910
P_1	19910	1750	1000	29910
P_2	29910	2356	1354	31214 14

(1). Allocation=MAX-Need. 找到的安全序列 $\{P_0, P_3, P_4, P_1, P_2\}$

(2). ① $Request_2(1,2,2,2) \leq Need_2(2,3,5,6)$

② $Request_2(1,2,2,2) \leq Available(1,6,22)$

Available = Availabe - Request = $(0,4,0,0)$

Allocation = Allocation + Request = $(2,5,7,6)$

Need = Need - Request = $(1,1,3,4)$.

Need 不满足小于 Available，不成立

(3) 系统立即满足 P_2 的请求$(1,2,2,2)$并没有马上进入死锁状态，

上述过程没有申请到新的资源，并因得不到资源进入阻塞态，只有当上述进程提出新的请求，导致所有没执行完的多个进程因得不到资源而阻塞并进入循环等待链，系统才进入死锁